

**METHOD FOR PROCESSING INFORMATION**

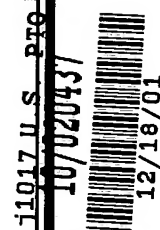
Patent Number: JP6139121  
Publication date: 1994-05-20  
Inventor(s): KANEKO KUNIYA  
Applicant(s):: TOYOTA MOTOR CORP  
Requested Patent: JP6139121  
Application Number: JP19920314171 19921028  
Priority Number(s):  
IPC Classification: G06F12/00  
EC Classification:  
Equivalents:

**Abstract**

**PURPOSE:**To provide an information processing method capable of repairing an I/O file generating a fault by minimum system constitution required and inspecting the cause of fault generation by knowing the contents of the I/O file held immediately before the generation of the fault.

**CONSTITUTION:**When the execution of a task 10 for accessing an I/O file 12 and executing file processing is instructed, the task ID of the task 10 whose execution is instructed and the file ID of the file 12 to be accessed are read out from a definition body file 16. A RAS task 14 is executed based upon the task ID and the file ID and the contents of the file 12 are stored in a RAS file 18 prior to the execution of the task 10. Thereby when a fault is generated at the time of executing the task 10, the file 12 can be repaired by copying the contents of the file 18 and the contents of the file 12 held immediately before the access can be known and used for inspecting the cause of the fault.

Data supplied from the esp@cenet database - I2



(19)日本国特許庁 (JP) (12) 公開特許公報 (A) (11)特許出願公開番号  
特開平6-139121  
(43)公開日 平成 6年(1994) 5月20日

(5)IntCl <sup>1</sup>	機別記号	庁内整理番号	FI	技術表示箇所
G 0 6 F 12/00	5 3 1 M	8626-5B		
審査請求 未請求 請求項の数 1 (全 9 頁)				

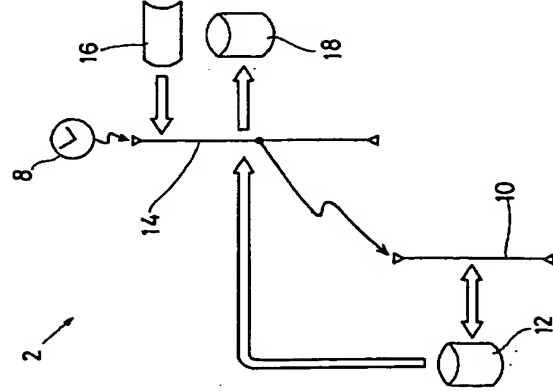
(21)出願番号	特開平4-314171	(71)出願人	000003207 トヨタ自動車株式会社 愛知県豊田市トヨタ町1番地
(22)出願日	平成 4年(1992)10月28日	(72)発明者	金子 邦也 愛知県豊田市トヨタ町1番地 トヨタ自動車株式会社内
		(74)代理人	弁理士 岡田 英彦 (外 2名)

(54)【発明の名称】 情報処理方法

(57)【要約】 (修正有)

【目的】 必要最小限のシステム構成で、障害が発生した入力ファイルの修復ができ、障害発生直前の入力ファイルの内容を知って障害発生原因の究明を行うことが可能な情報処理方法を提供する。

【構成】 入力ファイル12にアクセスしてファイル処理を行うタスク10の実行が指示されると、実行が指示されたタスク10のタスクIDとアクセスされる入力ファイル12のファイルIDが定義体ファイル16から読み取られる。このタスクID、ファイルIDに基づいてRASタスク14が実行され、入力ファイル12の内容が、タスク10の実行に先立ってRASファイル18へ格納される。従って、タスク10の実行時に障害が発生した場合には、RASファイル18の内容を復元すれば入力ファイル12を修復でき、またアクセス直前の入力ファイル12の内容を知って、障害の原因究明に役立てられる。



【特許請求の範囲】

【請求項1】 複数の入出力ファイルと、該複数の入出力ファイルのうちの特定の入出力ファイルにアクセスしてファイル処理を行うファイル処理タスクを実行する中央処理装置と、前記ファイル処理タスクのタスク識別子と該ファイル処理タスクがアクセスする入出力ファイルのファイル識別子とを組にして記憶している識別子ファイルと一時保存用ファイルとを有する情報処理システムのための情報処理方法であって、

前記ファイル処理タスクの起動指令が与えられたときに該起動指令に基づいて前記識別子ファイルを検索し、前記複数の入出力ファイルの中から前記ファイル処理タスクがアクセスする入出力ファイル特定する工程と、

特定された入出力ファイルのデータを前記一時保存用ファイルに転写する工程と、

特定された入出力ファイルに対して前記ファイル処理タスクを実行する工程と、

前記ファイル処理タスクの実行時に障害が発生したか否かを判定する判定工程と、

該判定工程で障害が発生したと判定された場合には前記一時保存用ファイルに基づく復旧処理を行う工程と、

前記判定工程で障害が発生していないと判定された場合には前記一時保存用ファイルの内容を消去する工程、とを有する情報処理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は、データファイル等の入出力ファイルにアクセスしてデータ更新処理等のファイル処理を行うタスクを実行する情報処理システムにおいて、ファイル処理を行うタスクの実行時に障害が生じた場合に、その原因の究明とファイルの回復を容易に行うことができる情報処理方法に関する。

【0002】

【従来の技術】 中央処理装置（以下、「CPU」とも略する。）を中心としてなるホストコンピュータ装置およびそれに付随する主記憶装置あるいは補助記憶装置と端末入出力装置等から構成される情報処理システムにおいて、データファイル等の入出力ファイルにアクセスしてデータ更新処理等のファイル処理を行うタスクを実行する際に、障害が発生する場合がある。このような場合に、障害発生原因の追求および障害が発生したファイルの回復を行う必要上から、情報処理システムに対するRAS機能の要求が大きくなってきた。RAS（Reliability, Availability, Serviceability）機能とは、タスクの実行時に障害が発生した場合に、その発生時刻、障害が発生したタスクの識別子（以下、「タスクID」とも略する。）、プログラムの行番号、読み書き（アクセス）していたファイルの識別子（以下、「ファイルID」とも略する。）等を時系列に記憶しておき、システム管理者等からの要求に応じて必要部分をプリンタ等の

出力装置に出力する機能である。

【0003】 このRAS機能について、図9を参照して説明する。図9は、従来の情報処理システムにおけるRAS機能の概略を示す概念図である。図9（A）はRAS機能の仕組みを示す概念図であり、図9（B）はRASファイルの内容の一例を示す図である。図9（A）に示されるように、ファイル処理を行う一般タスク110が起動されて、この一般タスク110の実行中に障害が発生した場合には、一般タスク110からRASタスク114に対して障害通知がされる。このRASタスク114は、インターバルタイマ108によって、定周期で起動されている。障害発生時の検知の方法としては、例えば、CPU（中央処理装置）のタイマ機能を用いて、一般タスク110の開始から一定時間経過しても復帰情報を受けないことにより障害を検知する方法等が用いられる。この障害通知によってRASタスク114が起動し、RASファイル118において障害処理が行われる。すなわち、図9（B）に示されるように、障害発生年月日と時刻、障害が発生したタスクのタスクID、プログラムの行番号およびアクセスしていたファイルのファイルIDが、RASファイル118に書き込まれる。

【0004】 しかしながら、このように障害処理が行われたRASファイル118から、障害が発生したファイルのファイルID（図9（B）の例では、「H124PF」）を読み取り、このファイルの内容を出力したとしても、障害発生原因が究明できない場合が多い。すなわち、このファイルの内容は、多くの場合アクセスによって変化を受けており、障害発生原因に最も密接に関連する障害発生直前のファイルの内容が失われてしまっているためである。このような場合には、RASファイル118を用いても障害発生原因の究明は困難となる。また、ファイルの内容が変化してしまっているために障害が発生したファイルの回復もできなくなる。そこで、このような場合にも障害が発生したファイルの回復を可能にするとともに、障害発生直前のファイルの内容を知ることでより障害発生原因の究明を行うための技術として、特開平4-18647号公報に記載された2元保存データ復旧処理方法とその機構の発明がされている。

【0005】 この公報に記載された技術について、図10を参照して説明する。図10に示されるように、この公報に記載の2元保存データ復旧処理方法に係る情報処理システム200は、中央処理装置204、識別子ファイル206、外部記憶装置212、主記憶装置216を有している。ここで、外部記憶装置212内には、複数の入出力ファイル214A、214B、…、214Zが保存されており、主記憶装置216内には、これと同数の入出力ファイル218A、218B、…、218Zが保存されている。そして、入出力ファイル214Aと2

イル18内に保存される。RASフアイル18内に保存される内容の一例を図5(B)に示す。図5(B)に示されるRASフアイル18に保存された内容は、入出力データフアイル12の内容と全く同一である。なお、RASフアイル18は、この図5(B)に示されるようなフアイル保存領域と、従来の図9(B)に示されるような障害処理領域とから成り立っている。

【0017】続いて、RASタスク14が、データ処理タスク10に起動をかける(図4のステップS16)。そしてデータ処理タスク10が作動して、入出力データフアイル12内のデータが読み出され、データ更新等の所定の処理が行われた後、再度入出力データフアイル12に書き込まれる(図4のステップS18)。次に、このデータ処理タスク10の進行時に障害が発生したか否かが、システム管理者等により判定される(図4のステップS20)。そして、障害が発生した場合には、システム管理者等の操作によりRASフアイル18の内容が読み出され、入出力データフアイル12に転写される(図4のステップS22)。これによって、障害が発生した入出力データフアイル12の内容を直ちに回復させることができる。また、アクセス直前の入出力データフアイル12の内容を知ることができ、システム管理者等による障害発生原因の究明が容易になる。一方、障害が発生しなかった場合には、所定の処理が行われた入出力データフアイル18の内容がそのまま保存される。不要になったRASフアイル18の内容は消去される(図4のステップS24)。なお、このRASフアイル18の消去は、実際には以下に述べるように、今回のRASフアイル18への書き込みの直前に自動的に実行される(図6のステップS50参照)。

【0018】次に、図2の入出力データフアイル12の内容をRASフアイル18へ書き込み処理(図4のステップS14)において行われる処理の手順について、図6を参照して詳細に説明する。図6は、本実施例の情報処理方法におけるフアイル内容のRASフアイルへの書き込み処理の手順を示すフローチャートである。まず、ステップS30において、以下に説明する書き込み処理のプログラムが起動される。続いて、ステップS32において、起動指令がされたデータ処理タスクがあるか否かが判定される。このステップS32における判定結果がNの、すなわち起動指令を受けたデータ処理タスクがない場合には、図6に示されるプログラムは待機状態となる。一方、ステップS32における判定結果がYの場合、すなわち起動指令を受けたデータ処理タスクがある場合には、ステップS34へ進んで、その起動指令を受けたデータ処理タスクのIDが受信される。

【0019】次に、定義体フアイル16が開かれて(ステップS36)、ステップS34で受信されたデータ処理タスクのタスクIDに基づいて、定義体フアイル16

に記憶された、当該データ処理タスクによって処理されるべき入出力データフアイルのIDが読み取られる(ステップS38)。その後、定義体フアイル16は閉じられる(ステップS40)。続いて、読み取られたIDに基づいて処理される入出力データフアイル12が開かれ(ステップS42)、開かれた入出力データフアイル12に記憶されたデータが読み取られる(ステップS44)。その後、入出力データフアイル12は閉じられる(ステップS46)。さらに、RASフアイル18が開かれ(ステップS48)、RASフアイル18の前の記憶内容がクリアされる(ステップS50)。そして、ステップS44で読み取られた入出力データフアイル12内のデータが、RASフアイル18に書き込まれる(ステップS52)。その後、RASフアイル18が閉じられ(ステップS54)、RASフアイルへの書き込み処理が終了する(ステップS56)。

#### 【0020】実施例2

次に、本発明を具現化した実施例2について、図7を参照して説明する。図7は、本発明に係る情報処理方法の実施例2を示す概念図である。図7に示されるように、本実施例の情報処理システム32では、タスクスケジューラ44(この中には、実施例1と同様のRASタスクが含まれる。)によってデータ処理タスク40が起動され、入出力データフアイル42内のデータが読み出されて処理された後に、再び入出力データフアイル42に書き込まれる。このデータ処理タスク40の実行に先立って、タスクスケジューラ44によって入出力データフアイル42内のデータが読み出され、RASフアイル48に保存される。そして、データ処理タスク40の実行に際してRAS出力タスク50が起動され、RASフアイル48内に保存された障害発生直前の入出力データフアイル42の内容が、デタリストとしてRAS出力プリンタ52に出力される。また、実施例1と同様に、システム管理者等の操作によって、RASフアイル48の内容が障害が発生した入出力データフアイル42に転写される。

【0021】このように、本実施例の情報処理方法は、アクセス直前の入出力データフアイル42の内容をRASフアイル48に保存することにより、障害が発生した場合の入出力データフアイル42の回復を可能にするとともに、障害の発生と同時にアクセス直前の入出力データフアイル42の内容が自動的に出力される。このRAS出力プリンタ52からの出力によって、システム管理者等は障害の発生を知るとともに、障害発生直前の入出力データフアイル42の内容を復元することが可能になり、マニュアル操作による入出力データフアイル42の修復とともに、障害発生原因の究明を行うことができる。

#### 【0022】実施例3

次に本発明を具現化した実施例3について、図8を参照して説明する。図8は、本発明に係る情報処理方法の実施例3を示す概念図である。図8に示されるように、本実施例の情報処理システム62においては、入出力データフアイル72に対してフアイル処理が行われる。まず、タスクスケジューラ74(この中には、実施例1と同様のRASタスクが含まれる。)によってデータ処理タスク70が起動され、入出力データフアイル72内のデータが読み出されて、このデータに対して所定の処理が行われる。そして、処理されたデータが再度入出力データフアイル72に書き込まれる。このデータ処理タスク70の実行に先立って、実施例1あるいは実施例2と同様に、タスクスケジューラ74によって入出力データフアイル72内のデータが読み出され、RASフアイル78に書き込まれる。そして、上記の入出力データフアイル72への書き込みの際に障害が発生した場合には、タスクスケジューラ74によってフアイル修復タスク80が起動される。このフアイル修復タスク80が実行されることによって、RASフアイル78内に保存された障害発生直前の入出力データフアイル72の内容が読み出されて、入出力データフアイル72に書き込まれる。【0023】このように、本実施例の情報処理方法においては、アクセスされる直前の入出力データフアイル72の内容をRASフアイル78に保存するだけでなく、障害が発生した場合には、このRASフアイル78に保存された内容によって入出力データフアイル72を自動的に修復する機能を持っている。これによって、情報処理システム62に障害が発生した場合には、システムの保守管理者等が作業を行うことなく、自動的に入出力データフアイル72の修復が行われ、極めて信頼性・保守性に優れた実用的な情報処理システムとなる。

【0024】上記の実施例においては、入出力フアイルとして、データの書き込みと読み出しを行う入出力データフアイルを例にとつて説明したが、この入出力フアイルはかかる入出力データフアイルでなくともよく、情報処理システムに接続された自動制御機械や自動ロボットのコントロルユニット等であってもよい。すなわち、本発明における入出力フアイルは、単なるデータ記憶装置だけでなく、自動制御機械のコントロールユニット内のメモリ装置等の、情報処理システムがアクセスして処理を行うことができる対象の全てを含むものである。情報処理システムのその他の部分の構成、大きさ、形状等や、情報処理方法のその他の工程についても、本実施例に限定されるものではない。

【0025】さらに、実施例2に固有の効果として、障害の発生と同時にアクセス直前の入出力データフアイルの内容が自動的に出力されるため、情報処理システムの保守管理者等が障害発生直前の入出力データフアイルの内容を復元することができ、障害発生原因の究明を行

うことができる。また、実施例3に固有の効果として、障害が発生した場合にはRASフアイルに保存された内容によって入出力データフアイルを自動的に修復する機能を有しているため、システム管理者等が作業を行うことなく自動的に入出力データフアイルの修復が行われるという利点がある。

#### 【0026】

【発明の効果】本発明においては、フアイル処理を行うタスクを実行する前にタスクがアクセスする入出力フアイルの内容を一時保存用フアイルへ転写しておき、障害が発生した場合には、一時保存用フアイルから入出力フアイルへ転写する情報処理方法を創出したために、必要最小限のシステム構成で、障害が発生した入出力フアイルの内容が回復されるとともに、障害発生直前の入出力フアイルの内容を知ることができ、これによって、情報処理システムの信頼性、可用性、保守性を著しく向上させることができる。極めて実用的な情報処理方法となる。

#### 【図面の簡単な説明】

【図1】本発明に係る情報処理方法の構成を模式的に示す図である。

【図2】本発明に係る情報処理方法の実施例1の基本的構成を示す概念図である。

【図3】情報処理方法の実施例1における情報処理システムの全体構成を示すブロック図である。

【図4】情報処理方法の実施例1における全体の流れを示すフローチャートである。

【図5】情報処理方法の実施例1におけるフアイル内容を示す図である。

【図6】情報処理方法の実施例1におけるフアイル内容のRASフアイルへの書き込み処理の手順を示すフローチャートである。

【図7】本発明に係る情報処理方法の実施例2を示す概念図である。

【図8】本発明に係る情報処理方法の実施例3を示す概念図である。

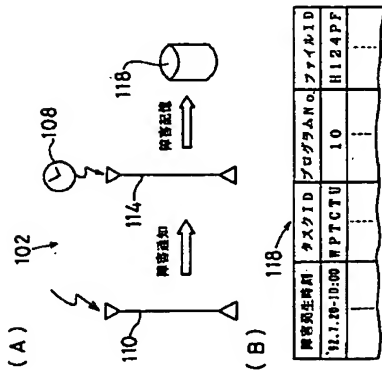
【図9】従来の情報処理システムにおけるRAS機能の概略を示す説明図である。

【図10】従来の情報処理システムにおける情報処理方法を示す図である。

#### 【符号の説明】

M0、2 情報処理システム  
M2、8 起動指令  
M4、18 一時保存用フアイル  
M6、20 中央処理装置  
M8、16 識別子フアイル  
M10、16A タスク識別子  
M12、16B フアイル識別子  
M14～M14、12 入出力フアイル

【図9】



【図10】

